Performance Tuning for Mac OS X Server

Session 620

These are confidential sessions—please refrain from streaming, blogging, or taking pictures.

Performance Tuning for Mac OS X Server

Steve Parker
Server Performance

What You'll Learn

- Current Snow Leopard Server performance
- Planning for good performance
- Building services for scale and performance
- Diagnosing performance problems

Snow Leopard Server Performance

Better than ever in Snow Leopard

Why We Benchmark

- Predict
- Provision
- Diagnose

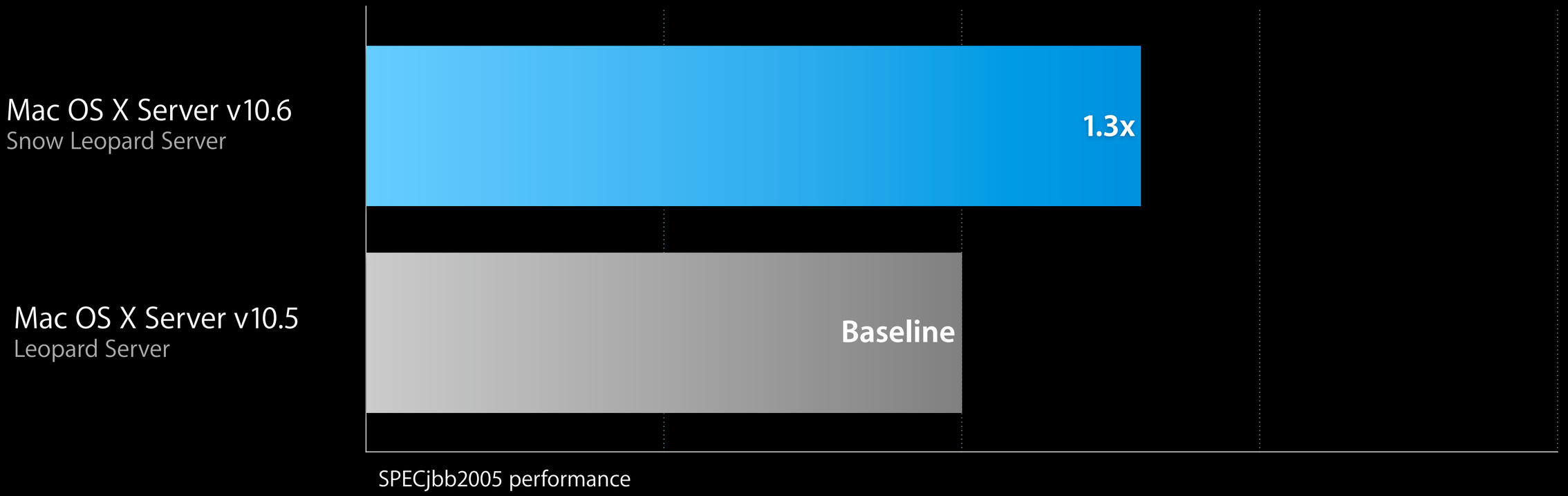
How Fast Can You Crunch Data?

SPECjbb2005

- Java application modeled on real application
  - Inventory and order entry system
  - In-memory tables
  - Scale up the number of warehouses inventoried
  - Find peak

Java Server Performance

SPECjbb2005



Testing conducted by Apple in May 2009 using a 2.93GHz 8-core Xserve (Early 2009) with 48GB RAM running a shipping version of Leopard Server v10.5.7 (SPECjbb2005 bops = 205,224, SPECjbb2005 bops/JVM = 51,306) and a prerelease version of Snow Leopard Server (SPECjbb2005 bops = 269,977, SPECjbb2005 bops/JVM = 67,494). SPEC® and SPECjbb2005® are trademarks of the Standard Performance Evaluation Corporation (SPEC); see www.spec.org for more information. Competitive benchmark results stated here reflect internal Apple testing and were submitted to SPEC in May 2009. For the latest SPECjbb2005 benchmark results, visit www.spec.org/osg/jbb2005. Performance tests are conducted using specific computer systems and reflect the approximate performance of Xserve and Mac OS X Server.

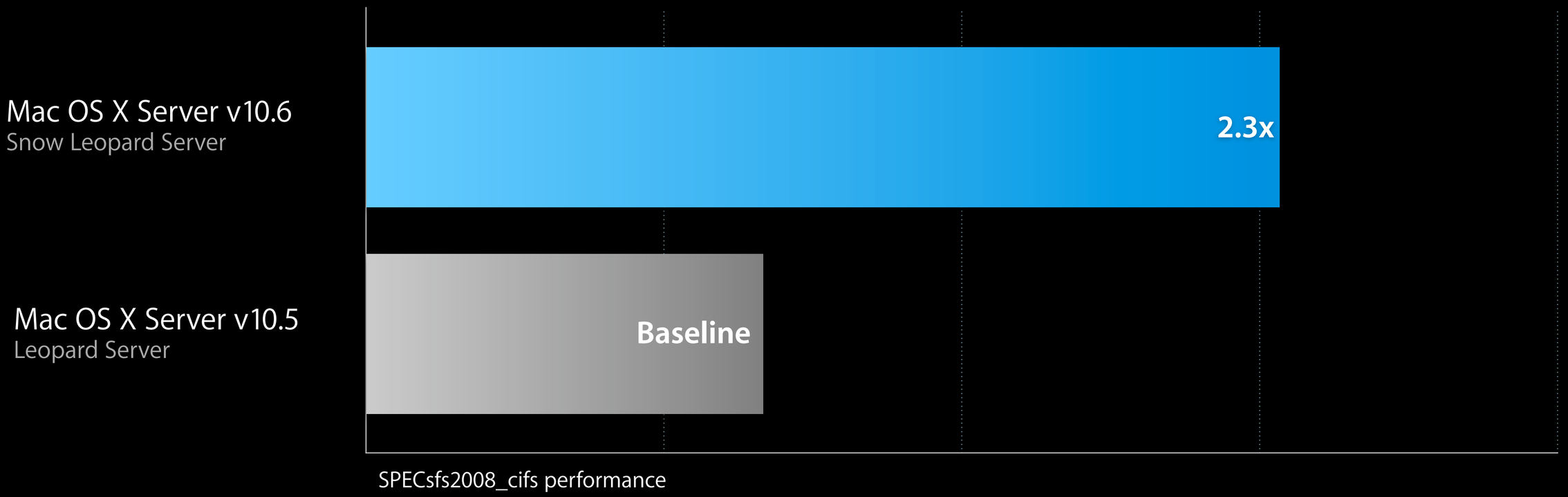
How Fast Is Your File Server?

SPECsfs2008

- Simulate NFS and CIFS Clients
  - Parallel clients, randomized request timing
  - 60% metadata/18% read/10% write
  - 120MB/sec per client per op/sec
  - Space grows by 10%
  - Synthetic clients

File Server Performance (SMB)

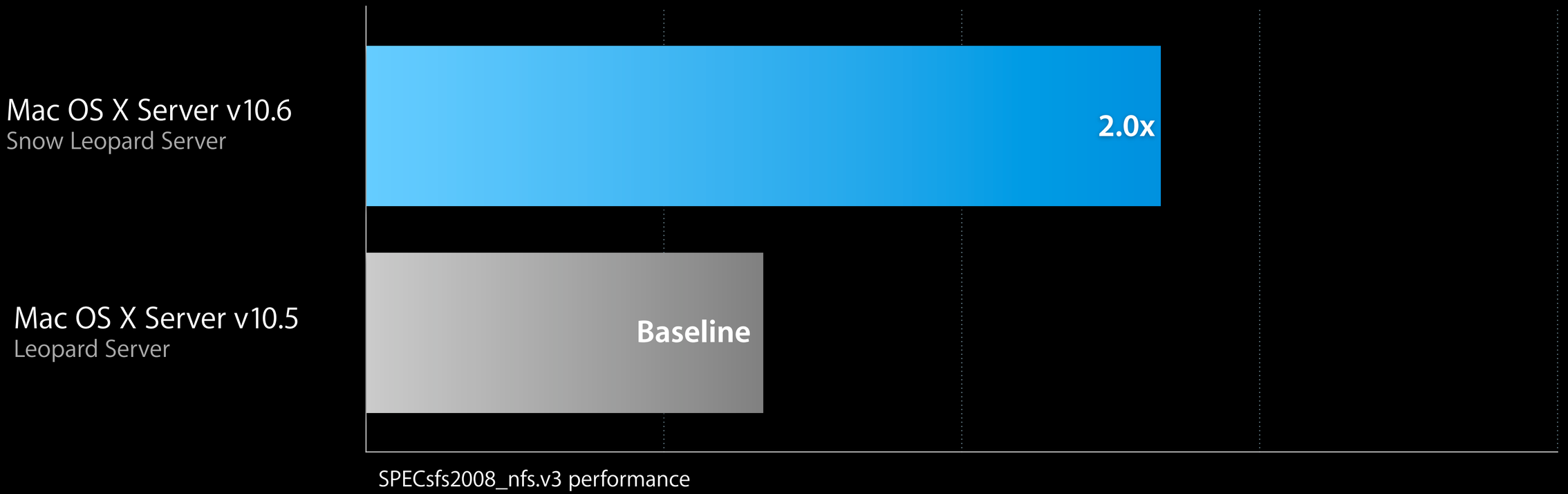
SPECsfs2008_cifs



Testing conducted by Apple in May 2009 using a 2.93GHz 8-core Xserve (Early 2009) with 48GB RAM, a 4-port 4-Gbps Fibre Channel adapter, a SmallTree 6-port Gigabit Ethernet adapter, and 4 Promise VTrak E-class RAID arrays, each with 16 300GB 15,000-rpm SAS drives, running a shipping version of Leopard Server v10.5.7 and a prerelease version of Snow Leopard Server. Testing with Leopard Server v10.5.7 reported 19,313 SPECsfs2008_cifs ops per second with an overall response time of 1.89 ms, and testing with Snow Leopard Server reported 44,347 SPECsfs2008_cifs ops per second with an overall response time of 1.89 ms. SPEC® and the benchmark name SPECsfs2008™ are trademarks of the Standard Performance Evaluation Corporation (SPEC); see www.spec.org for more information. Competitive benchmark results stated here reflect internal Apple testing and were submitted to SPEC in May 2009. Performance tests are conducted using specific computer systems and reflect the approximate performance of Xserve and Mac OS X Server.

File Server Performance (NFS)

SPECsfs2008_nfs.v3



Testing conducted by Apple in May 2009 using a 2.93GHz 8-core Xserve (Early 2009) with 48GB RAM, a 4-port 4-Gbps Fibre Channel adapter, a SmallTree 6-port Gigabit Ethernet adapter, and 4 Promise VTrak E-class RAID arrays, each with 16 300GB 15,000-rpm SAS drives, running a shipping version of Leopard Server v10.5.7 and a prerelease version of Snow Leopard Server. Testing with Leopard Server v10.5.7 reported 9189 SPECsfs2008_nfs.v3 ops per second with an overall response time of 2.18 ms, and testing with Snow Leopard Server reported 18,784 SPECsfs2008_nfs.v3 ops per second with an overall response time of 2.67 ms. SPEC® and the benchmark name SPECsfs2008™ are trademarks of the Standard Performance Evaluation Corporation (SPEC); see www.spec.org for more information. Competitive benchmark results stated here reflect internal Apple testing and were submitted to SPEC in May 2009. Performance tests are conducted using specific computer systems and reflect the approximate performance of Xserve and Mac OS X Server.

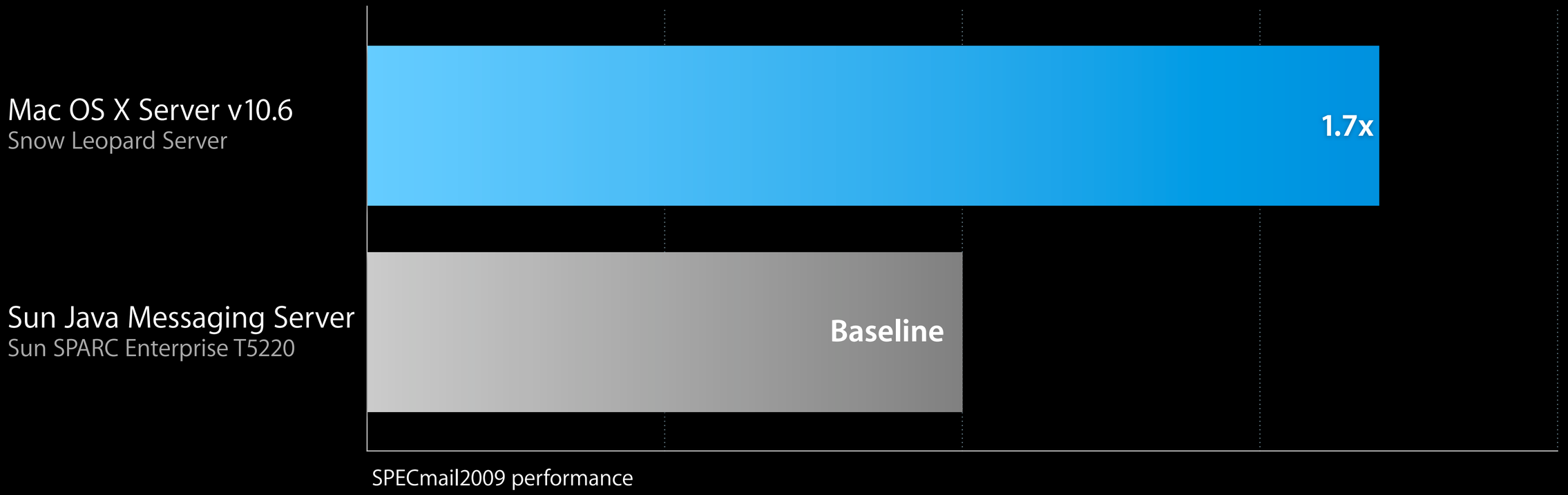
How Fast Is Your Mail Server?

SPECmail_Ent2009

- Simulate IMAP Clients
  - Many ways to do a given operation
  - Every implementation is different
  - Track message contents
  - Folders and heirarchy
  - Message size distribution

Mail Server Performance

SPECmail_Ent2009



Testing conducted by Apple in May 2009 using a 2.93GHz 8-core Xserve (Early 2009) with 48GB RAM, a 4-port 4-Gbps Fibre Channel adapter, a SmallTree 6-port Gigabit Ethernet adapter, and 5 Promise VTrak E-class RAID arrays, each with 16 300GB 15,000-rpm SAS drives, running a prerelease version of Snow Leopard Server. Apple testing produced results of 28,887 SPECmail_Ent2009 sessions per hour. Testing conducted by Sun Microsystems on a Sun SPARC Enterprise T5220 produced results of 17,316 SPECmail_Ent2009 sessions per hour. SPEC® and the benchmark name SPECmail2009™ are trademarks of the Standard Performance Evaluation Corporation (SPEC); see www.spec.org for more information. For the latest SPECmail2009 results, visit www.spec.org/osg/mail2009. Benchmark results stated here reflect internal Apple and Sun testing and were published on www.spec.org from March 2009 to June 2009. Performance tests are conducted using specific computer systems and reflect the approximate performance of Xserve and Mac OS X Server.

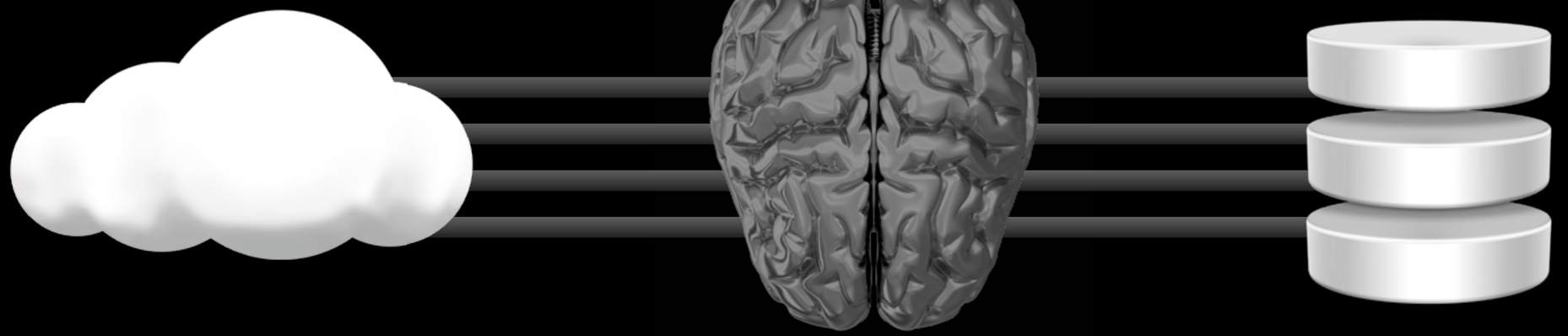
Planning for Good Performance

How to make sure your system performs at its best

SPEC vs. Reality

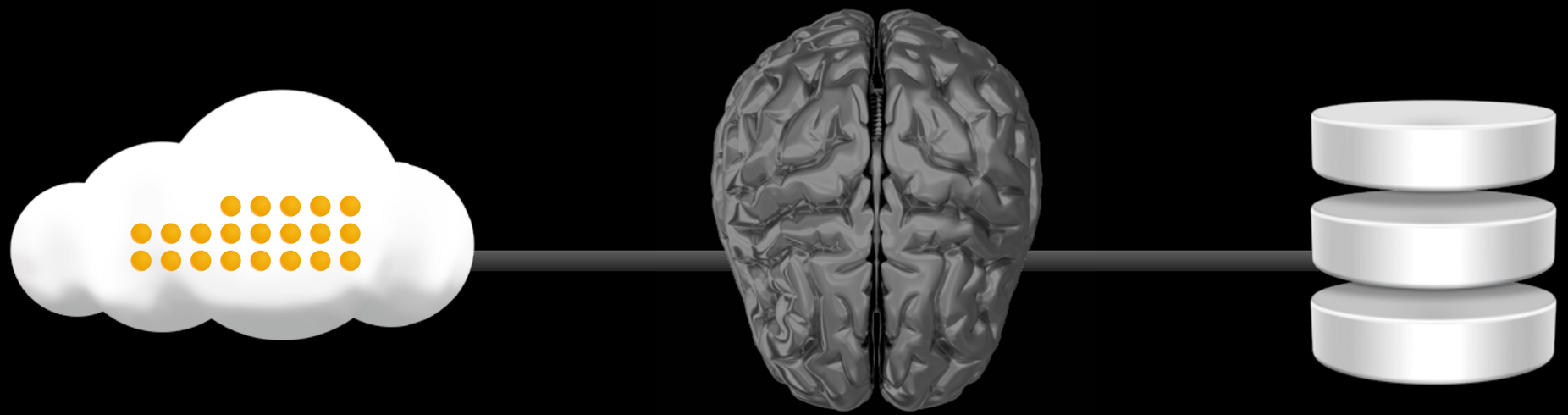
- Have you ever found that you didn’t realize the gain a benchmark predicted?
- How much faster will my system be?
- Depends on end-to-end system performance
  - Is the plumbing all big enough?

Big Brains and Small Pipes



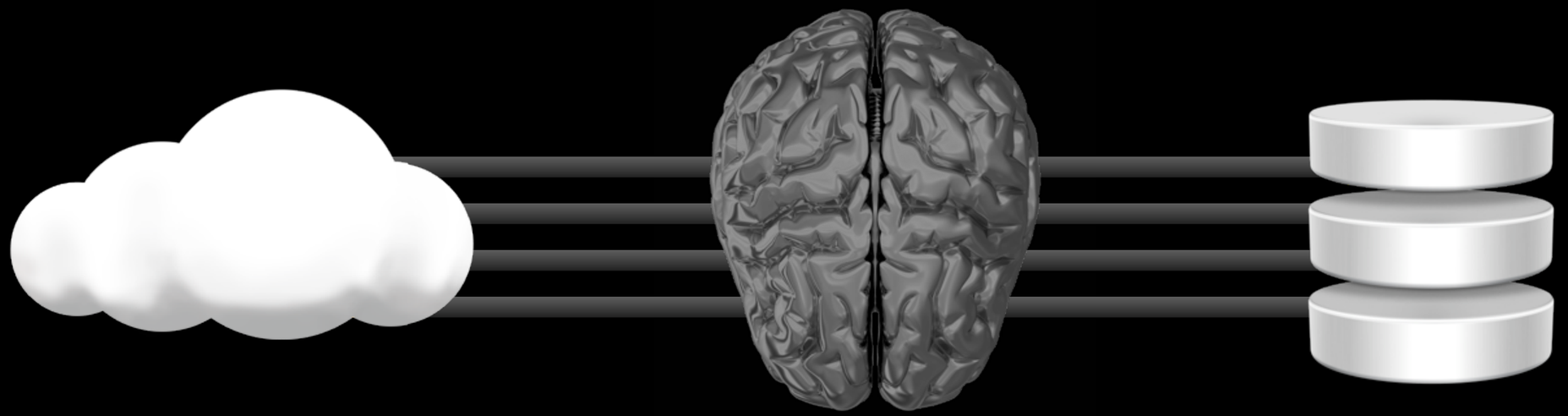
The Problem:
Balance throughput so that disk and network resources are available to ensure smooth flow of data

Big Brains and Small Pipes



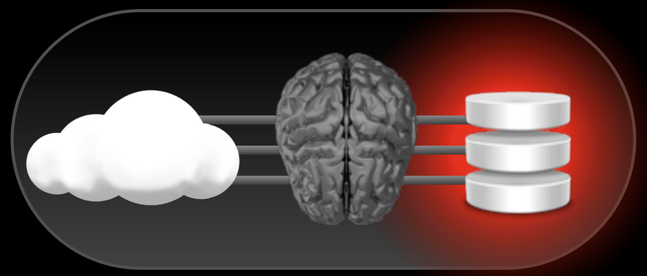
The Problem:
Balance throughput so that disk and network resources are available to ensure smooth flow of data

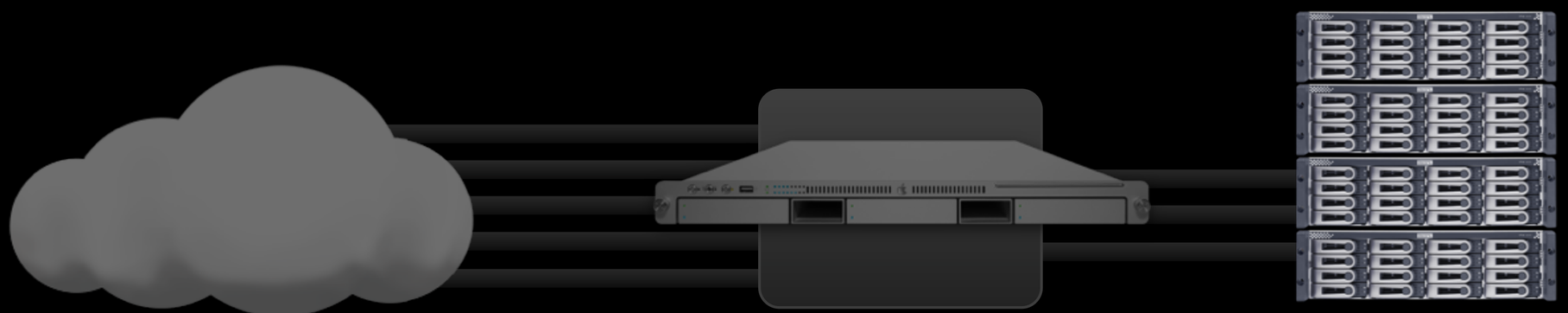
Big Brains and Small Pipes



The Problem:
Balance throughput so that disk and network resources are available to ensure smooth flow of data

Example: NFS





Published score is not peak
40% idle CPU
Need lots more disks…

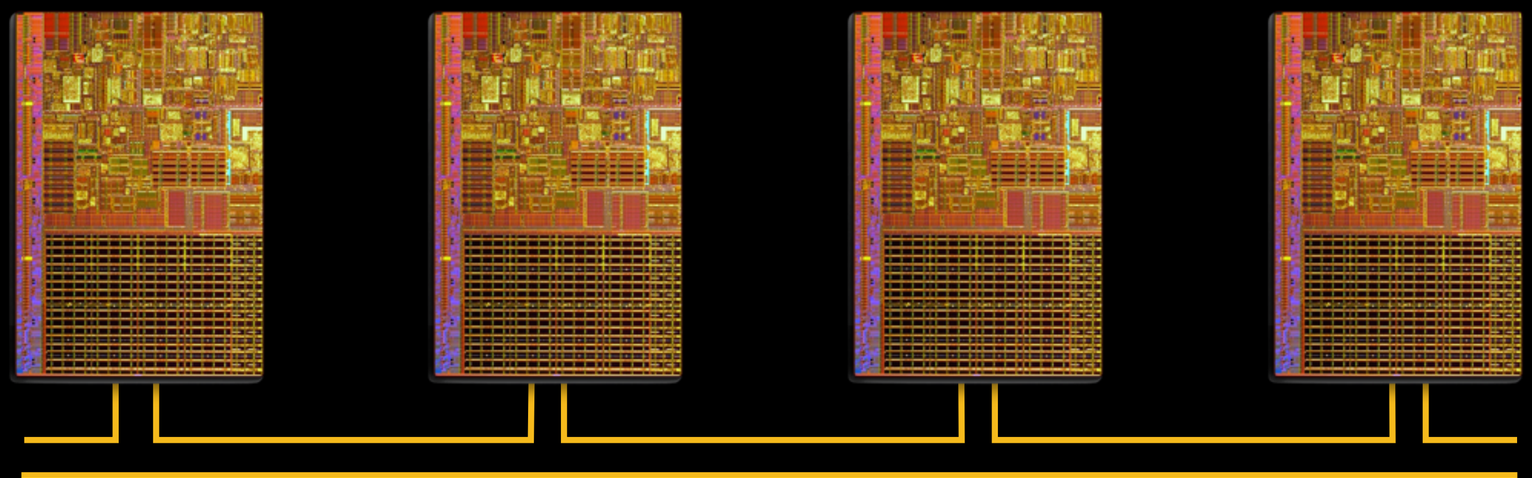
SPEC Postings—"Speed of Light"

- Four Promise VTrak E-class SAS arrays
  - 64 drives
  - 8MB of battery back RAM
  - 16Gbps of Fibre Channel interconnect
- Approximate cost: $80K
- 32 Separate HFS+ File systems
- *ALL of these are necessary to provide this class of NFS service*

Why Is This Good News?

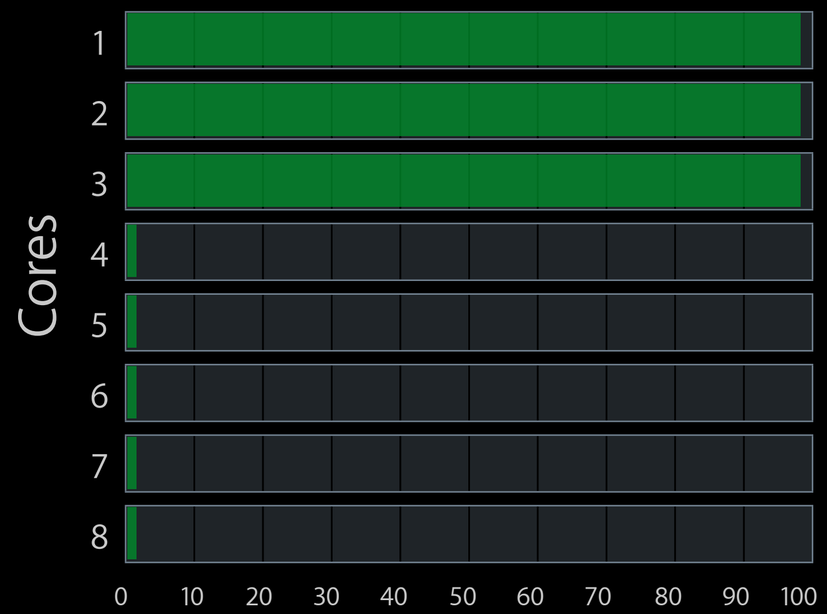
Server scales

- Threads operate separately and fully independently
- Paths handle large volumes of traffic
  - 8-NICs, can operate in parallel
  - 840MB/sec to the disk, random I/O pattern

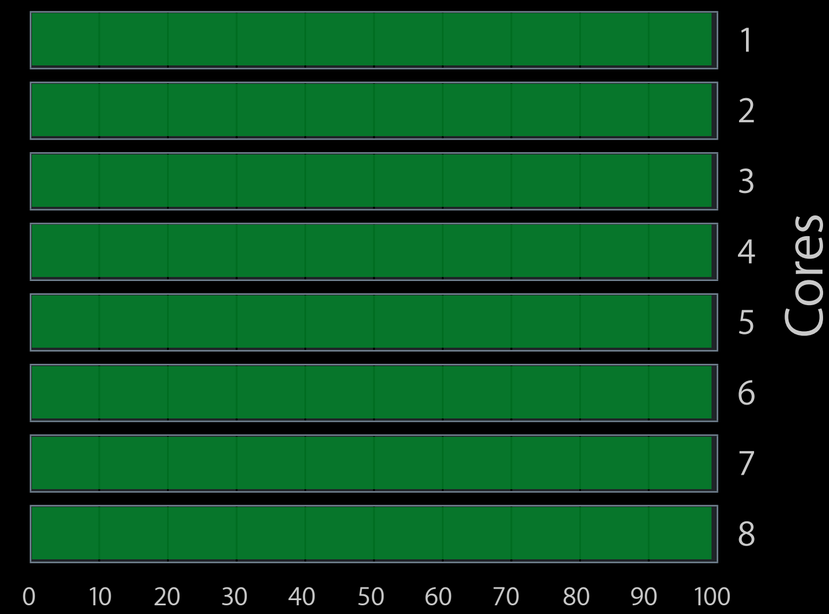


Multicore Scalability

Leopard Server

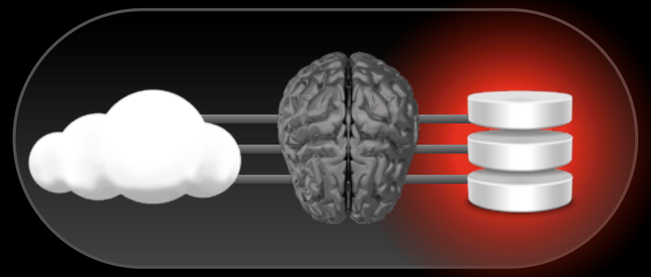


Snow Leopard Server



Snow Leopard Server Scales to utilize all cores, independently

Disk Performance Matters

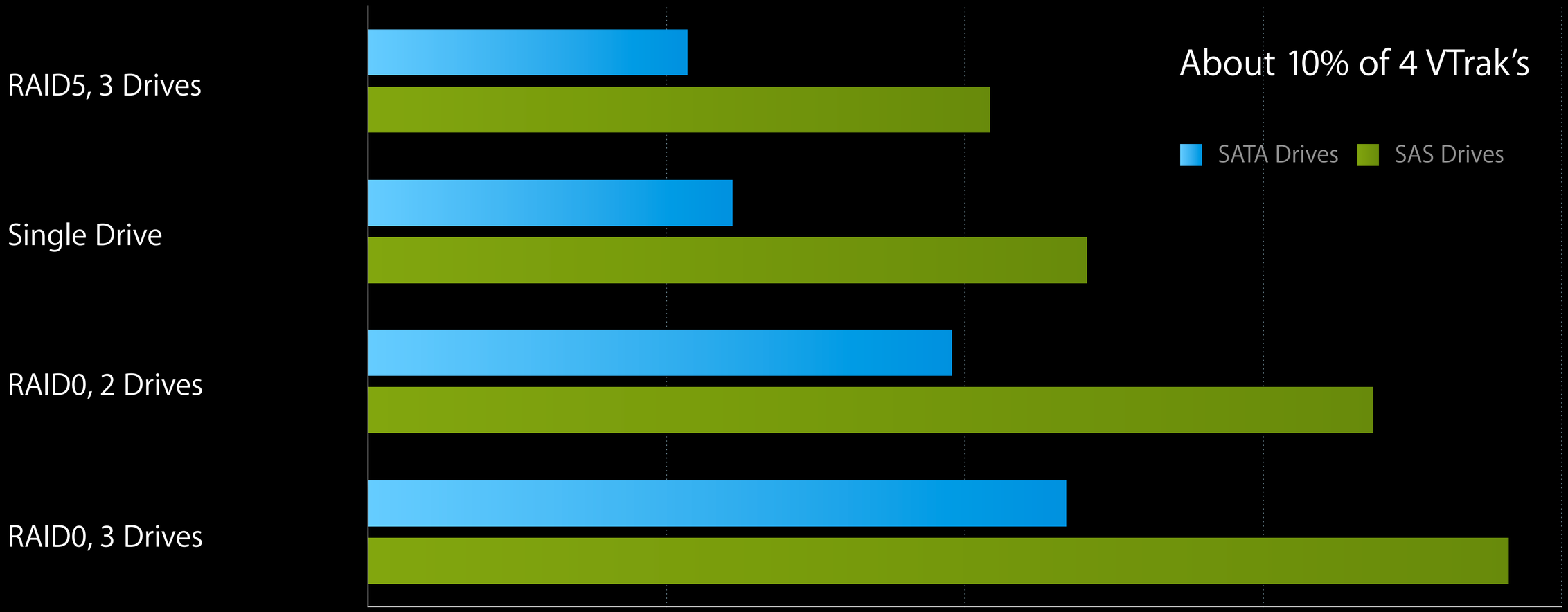


- Example: Drop any HFS+ Volumes or drives, drops the score
- Same amount of controllers
- Same amount of battery-backed RAM

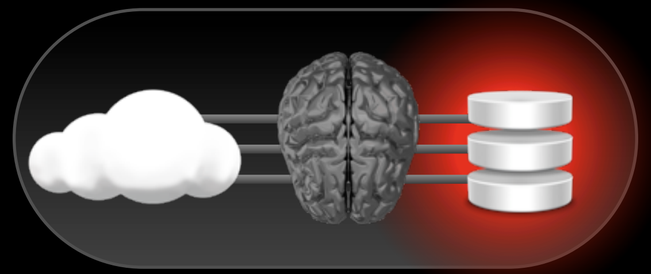
Your Performance—Your Disk Subsystem

- Example: Single internal SATA drive
  - 1/40th of the published score
  - $200—1/400th the cost

Xserve Drive Options



Disk Technologies



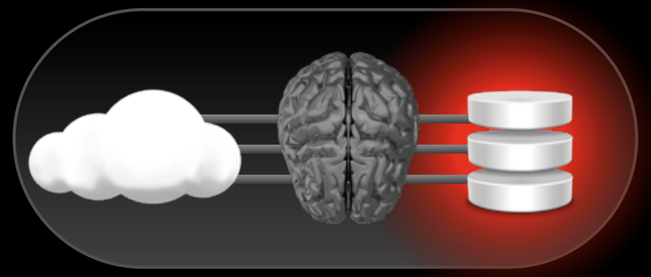
	Speed	Capacity	Application
SATA drive modules	Slow	3 TB	Medium size data sets, modest I/O speeds
SAS drive modules	Medium	.9 TB	Small data sets, medium I/O needs
Xserve H/W RAID	Improvement based on drive module	As above	Improved speed, plus off-loaded RAID processing
VTrak SATA RAID	Slightly better than H/W RAID and three-drive SAS	12 TB	Large data sets, medium I/O needs Best with relatively sequential or large I/O size workloads
VTrak SAS RAID	Fast	4.8 TB	Large data sets, high I/O speeds For example, large home directory server

Metadata Rate of Change

Most challenging aspect of server file system performance

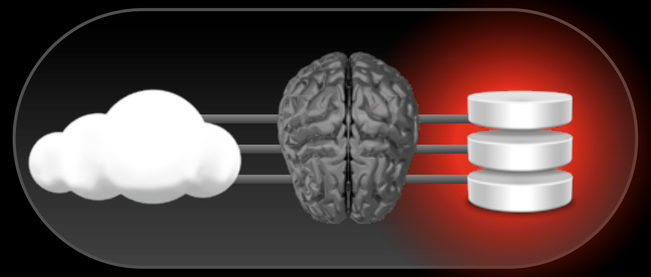
- Operations:
  - Create a file or directory
  - Remove a file or directory
  - Rename a file or directory
  - Change ownership, permissions or attributes
- On HFS+:
  - Rebalance the B-tree
  - Commit changes to the journal
  - Update the catalog file

Design Your Volumes



	HFS+	Xsan
Efficient Use of Space	Excellent	Fair
Caching	Scales with RAM	Localized by client
Clustering	None	Yes
Metadata Updates	Poor	Fair
Data Writes	Good	Outstanding

Workload Characteristics



	Disk Load	Key	Planning Priority	Best Choice
Mail	High	Fast, adequate disks	High Metadata update rate	Fibre Channel Multiple HFS+ Volumes
Web	Low	RAM cacheing	Plan for non-disk concerns: RAM, number of processes	Built-in SATA
Calendar Server	Moderate	Metadata cache	K64, large RAM	Built-in SATA or SAS
Home Directory Shares	High	Network and overall capacity	Can the network fabric deliver? Only moderate caching, so I/O rate important	Fibre Channel Multiple HFS+ Volumes
Podcast Producer	Moderate spped, High Capacity	Outstanding	Good	Xsan
Wiki Server	Low	RAM Cacheing	Adequate Networking Meet the necessary disk update rate	Any

Building for Scale and Performance

Creating services that are big and fast

Making Applications Fast
Building in Speed and Scale

- kqueue()
- Plan resource Limits
- listen() queue depth

Choose kqueue()

- select() and poll() both degrade nonlinearly with open files
- kqueue()—easy to use, notification based model
  - Always the most efficient choice
  - Create the file descriptor: kq = kqueue();
  - Get the next event: nevents = kevent(...);
  - Includes file descriptor and an additional pointer

Resource Limits

- launchd.plist(5)
- Protect the system from runaways
- Set properly for your application:
  - Maximum CPU time
  - Maximum data segment size
  - Largest file size
  - Open files at peak
  - Resident set size

listen() Queue Depth

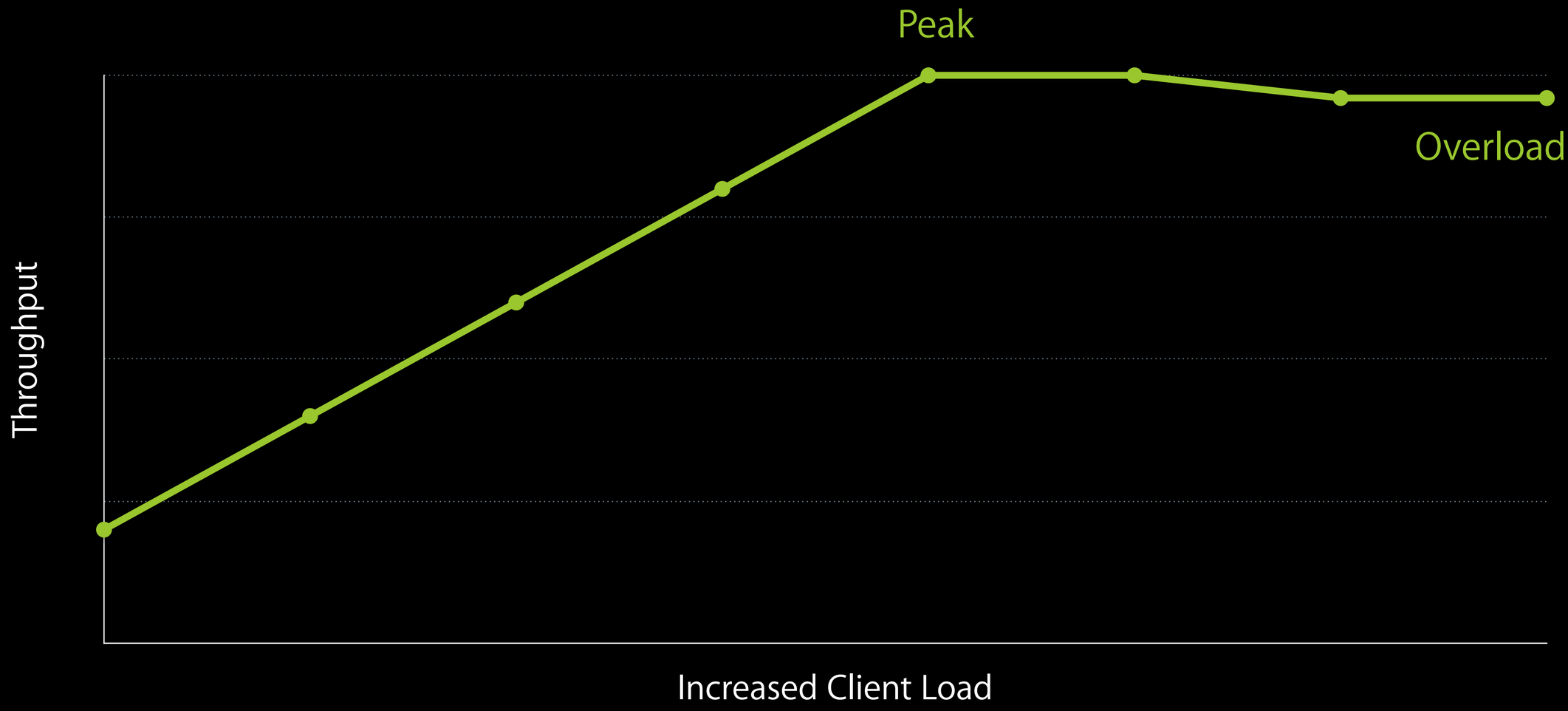
- Ensure spikes in connection establishment dealt with smoothly
- Don’t drop a connection just because there’s no thread or process available now
- We might be spawning the new process now

Diagnosing Performance Problems

When you haven't hit the speed of light, find out why

Empirical Performance Evaluation

Peak Throughput

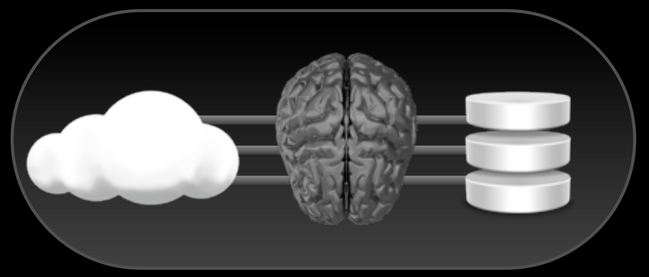


Empirical Performance Evaluation

Latency

top-guide.pl

How fast is data flowing?



Time:	CPU usage: user	sys	Idle	300-s avg.	Reads/sec: number	size	Writes/sec: number	size	Net in/sec: pkts	bits	Net out/sec: pkts	bits	kernel CPU%
07:04:42	3.2	1.9	94.8	82.0	84	338K	11	373K	506	1899K	437	298K	1.9
07:04:52	2.9	1.3	95.8	86.6	71	291K	0	302K	458	2123K	370	246K	1.5
07:05:02	2.5	1.6	95.9	88.9	55	220K	111	838K	417	1912K	340	226K	1.6
07:05:12	2.6	1.8	95.6	90.3	88	353K	3	290K	417	2157K	328	220K	1.6
07:05:22	3.2	1.6	95.2	91.1	85	345K	0	239K	614	3013K	495	328K	1.9
07:05:33	4.6	2.4	93.0	91.4	114	461K	121	1346K	834	3952K	687	453K	2.6
07:05:43	5.3	2.8	92.0	91.4	163	654K	16	671K	909	3920K	765	512K	2.9
07:05:53	5.2	2.2	92.7	91.6	132	532K	0	409K	905	4311K	737	487K	2.5
07:06:03	4.6	2.3	93.1	91.7	92	372K	173	1602K	740	3730K	595	391K	2.4
07:06:13	5.8	2.5	91.7	91.7	126	512K	8	518K	887	4117K	733	487K	2.7
07:06:23	5.5	2.4	92.1	91.8	157	630K	6	490K	972	4355K	815	543K	2.8
07:06:33	5.2	2.6	92.3	91.8	170	685K	76	1139K	869	4180K	715	472K	2.8
07:06:43	4.2	2.4	93.4	91.9	120	482K	112	711K	680	3232K	561	372K	2.4
07:06:53	5.3	2.1	92.6	92.0	132	535K	2	838K	941	4288K	779	515K	2.7
07:07:03	5.4	2.4	92.3	92.0	167	671K	15	666K	832	3418K	712	477K	2.7
07:07:13	4.7	2.6	92.7	92.0	92	378K	176	1019K	693	3262K	569	379K	2.4
07:07:23	5.8	2.2	92.0	92.0	128	512K	3	834K	1039	5364K	834	548K	2.7
07:07:33	5.9	2.4	91.8	92.0	131	529K	10	625K	1008	5087K	817	536K	2.8
07:07:43	4.2	2.6	93.1	92.1	107	430K	175	1537K	681	3338K	554	365K	2.4

Study Throughput vs. Load

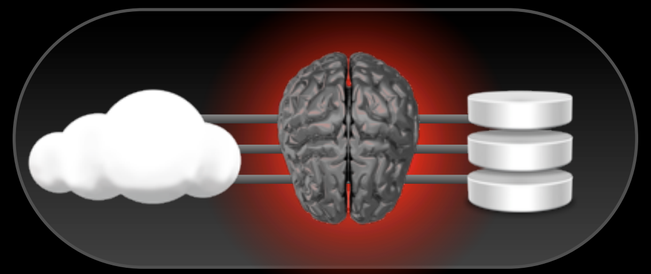
- Review the increase in usage versus load
  - Linear is healthy—more capacity, put to use
  - Super linear is the sign of a bottleneck
- Examples:
  - 60% of CPU services 200 clients
  - 60%/200 = .3%
  - Add 50 clients, measure 86% CPU usage: 86%/250 = .34% each
  - Investigate for lock-thrashing

General Starting Point

Kbase HT1992

- CPU
  - Use dtrace to verify no mutex thrashing
    - lockstat:::adaptive-block
    - lockstat:::rw-block
    - Multithread applications: plockstat -C
- Network traffic
  - Packet counts and bandwidth
- Disk I/O
  - Harder to measure directly

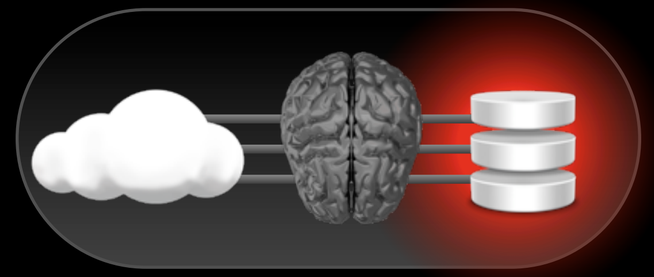
Interpreting plockstat -C



```
Mutex block

Count     nsec Lock                         Caller
-------------------------------------------------------------------
 2027    22281 0x805eb0                     a.out`process_request+0x3d1
 1637    25794 0x805eb0                     a.out`release_block+0xac
 ....

R/W reader block

Count     nsec Lock                         Caller
-------------------------------------------------------------------
    8   762506 0xa6d9e8                     a.out`key_find+0x121

R/W writer block

Count     nsec Lock                         Caller
-------------------------------------------------------------------
    3  2916833 0xa6d9e8                     a.out`update_key_table+0x21d
```

Disk I/O

HFS+ journal update rate

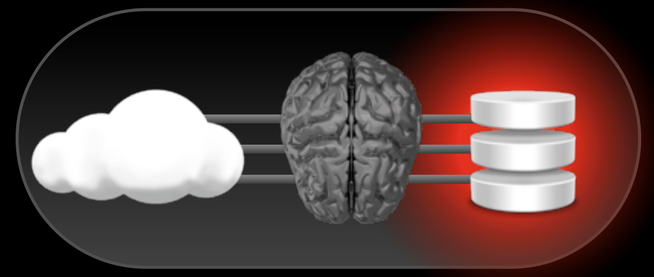


- Disk I/O
  - Determine journaling performance limit with DTrace
  - mtxheat.d script

```
Volume  354027080
              mach_kernel`lck_mtx_lock_wait_x86+0x1c7
              mach_kernel`lck_mtx_lock+0x217
              mach_kernel`journal_start_transaction+0x21c
             179
```

Journal I/O rate per second

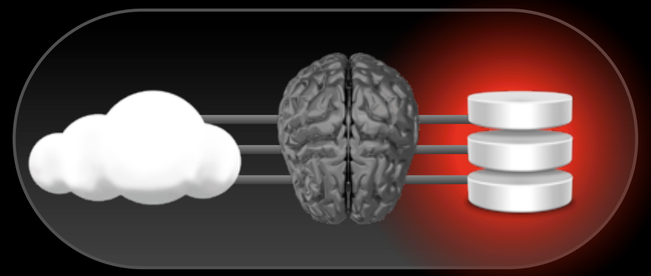
Per Volume Disk Usage



- iopending(1m)
- How deep is the disk queue?
- Focus on one filesystem with “-m <volname>”

```
root# iopending -m Data
Tracing... Please wait.
2008 Jun  1 23:24:05,  load: 24.74,  disk_r:      4 KB,  disk_w: 155208 KB

          value  ------------- Distribution ------------- count
            < 0 |                                         0
              0 |@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@@ 31387
              1 |@@@@@                                    5272
              2 |@                                        1392
              3 |@@                                       1723
              4 |                                         229
              5 |                                         8
              6 |                                         0
```

Evaluating Spotlight Impact



- Index maintained on the fly
- Under write-heavy loads may be high cost
  - Watch for CPU load from mds using ps or top
  - Control system-wide by:
    - `service com.apple.metadata.mds stop`
    - `service com.apple.metadata.mds start`
  - Disable per-volume: `mdutil -i off /Volumes/MyVolume`
  - Disable per-node: `touch name.noindex`

More Information

Mark Malone
Internet and Server Technologies Evangelist
mmalone@apple.com

MacOS X Server 10.5: Evaluating Performance
http://support.apple.com/kb/HT1992

How to Use DTrace
http://www.sun.com/software/solaris/howtoguides/dtracehowto.jsp

Apple Training and Certification
http://training.apple.com

Related Sessions

Advanced Debugging and Performance Analysis	Nob Hill Thursday 9:00AM
Mac OS X Server State of the Union	Pacific Heights Monday 5:00PM
Managing Processes With launchd	Russian Hill Friday 2:00PM
Performance Tuning with Shark on Mac and iPhone	North Beach Friday 9:00AM

Labs

Leopard Server Performance Lab	Information Technologies Lab Thursday 3:30PM
Advanced Debugging and Performance Analysis	OS X Lab B Thursday 9:00AM
64-bit Kernel Lab	Snow Leopard Lab Wednesday 12:00PM
Networking Lab	OS X Lab D Wednesday 2:00PM
Networking Lab	OS X Lab B Thursday 2:00PM
Performance Tuning with Shark on Mac and iPhone Lab	iPhone Lab C Friday 2:00PM

Summary

- Good performance depends providing a planned path thru network, CPU, and disk
- Plan for your workload and evaluate by measurement

Q&A

WWDC Session Survey

You can rate your sessions on the WWDC Attendee Site.

Survey forms are located on each session description page at:

developer.apple.com/wwdc/attendee

WWDC09